laboratorium: zadanie 10 termin: 22–26 maja 2023 r.

KURS JĘZYKA C++

MANIPULATORY I PLIKI

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

Prolog.

Dane przetwarzane przez komputery mogą być cyfrową reprezentacją dowolnych informacji: tekstowych, obrazowych, dźwiękowych, filmowych itp. Dane są pamiętane w postaci *plików* na nośnikach elektronicznych. Plik to uporządkowana sekwencja danych o skończonej długości. Dane w plikach mogą mieć reprezentację tekstową (czytelną i łatwą w edycji), albo binarną (przystosowaną do bezpośredniego przetwarzania przez procesor). Tak czy inaczej plik jest podstawową jednostką danych w systemie plików.

Zadanie 1.

Dla tekstowego strumienia wejściowego `istream` zdefiniuj własny manipulator bezparametrowy `clearline`, który będzie usuwał wszystkie znaki, aż do napotkania znaku przejścia do nowej linii (ten znak także należy usunąć ze strumienia) lub znaku końca pliku. Zdefiniuj również manipulator z parametrem `ignore (int x)`, którego zadaniem będzie pominięcie `x` znaków ze strumienia wejściowego, chyba że wcześniej zostanie wyjęty znak przejścia do nowej linii lub stumień się skończy.

Dla tekstowego strumienia wyjściowego `ostream` zdefiniuj bezparametrowe manipulatory `comma` wypisujący przecinek z odstępem `", "` oraz `colon` wypisujący dwukropek z odstępem `": "`. Zdefiniuj także manipulator z parametrem `index (int x, int w)`, który wypisze liczbę `x` w nawiasach kwadratowych i na liczbę tą przeznaczy co najmniej `w` pozycji (dosuń liczbę do prawego nawiasu kwadratowego).

Napisz program testujący zdefiniowane przez ciebie manipulatory. Jednym z elementów testowania ma być procedura, która odczyta wszystkie linie danych zapamiętując je w kontenerze `vector<>`, następnie posortuje odczytane linie leksykograficznie i wypisze je wraz z pierwotnymi numerami linii (numer linii umieść na początku wiersza w nawiasach kwadratowych).

Zadanie 2.

W oparciu o technikę *zdobywania zasobów poprzez inicjalizację* zaimplementuj bezpieczne klasy opakowujące pliki: `wejscie` dla plików binarnych do czytania (opakowanie dla obiektu `ifstream`) oraz `wyjscie` dla plików binarnych do pisania (opakowanie dla obiektu `ofstream`). Plik należy otworzyć w konstruktorze (jeśli okaże się to nieżmożliwe zgłoś wyjątek) a zamknąć w destruktorze. Zadbaj, by ustawienie flagi błędu `ios_base::badbit` lub `ios_base::failbit` powodowało automatyczne zgłoszenie wyjątku `ios_base::failure`.

Klasa `wejscie` powinna umieć odczytać dane binarne różnych typów podstawowych `char`, `int`, `double` itd., odczytując z pliku i umieszczając w pamięci odpowiednią liczbę bajtów (taką jaką wymaga wartość określonego typu). Natomiast klasa `wyjscie` powinna umieć zapisać dane binarne różnych typów podstawowych `char`, `int`, `double` itd., zapisując do pliku odpowiednią liczbę bajtów. Do realizacji operacji czytania oraz pisania danych różnych typów podstawowych zdefiniuj szablony funkcji składowych, które w zależności od typu danych będą odpowiednio odczytywać albo zapisywać odpowiednią liczbę bajtów.

Napisz program testujący zdefiniowane przez ciebie klasy. Jednym z elementów testowania ma być procedura, która zapisze do pliku kilka losowych liczb całkowitych a potem odczyta je z tego pliku (dane zapisywane a potem odczytywane z pliku wypisuj w postaci tekstowej na konsoli). Drugim elementem testowania ma być odczytanie pliku bajt po bajcie i wypisanie na konsolę odczytywanych wartości w postaci znaków (tylko te reprezentowane widoczne na klawiaturze), liczb szesnastkowych i całkowitych (wartości od 0 do 255).

Elementy w programie, na które należy zwracać uwagę.

- Podział programu na pliki nagłówkowe i źródłowe.
- Definicja manipulatorów bezparametrowych.
- Definicja manipulatorów z parametrami.
- Testowanie manipulatorów.
- Wrappery dla plików binarnych realizujące ideę RAI (zdobywanie zasobów poprzez inicjalizację).
- Szablony funkcji składowych realizujących zapis (w pliku do wyjściowym) i odczyt (z pliku wejściowego) danych w postaci binarnej.
- Obsługa błędów w strumianiach za pomocą wyjątków.
- Testowanie wrapperów strumieniowych.